ENHANCING YELP RATING PREDICTIONS

CS 182 FINAL PROJECT

Charles Hong
U.C. Berkeley
charleshong@berkeley.edu

Kevin Zhu
U.C. Berkeley
kevin_zhu@berkeley.edu

November 5, 2021

ABSTRACT

Natural Language Processing (NLP) is a field of growing importance, and is often used to analyze large quantities of user-generated text. In this project, we explore and build upon deep learning methods to predict the star rating of Yelp reviews. Specifically, we evaluate different approaches to enhancing the popular transformer model Bidirectional Encoder Representations from Transformers (BERT). Our metrics for evaluation are accuracy, whether the model correctly predicts a review's 1–5 star rating, and average star error, the numerical distance between the model's prediction and the correct star value. To address this problem, we introduce novel ensembling methods that are intended to improve the generalizability of BERT by reducing variance while fitting to both objectives. Specifically, our experiments conclude that through the use of the methods we devise, we can improve accuracy by 3.5% and mean absolute error by 27% on this task relative to a base BERT model fine-tuned on task-specific data.

1 Introduction

Today, Internet users often turn to user-generated review sites such as Yelp or IMDb to guide their shopping and entertainment choices. In turn, such businesses analyze user-uploaded text using various NLP approaches, in order to optimize their algorithms in areas such as recommendation or search. In particular, companies are increasingly using Sentiment Analysis to classify opinions and sentiments expressed in text [1]. These techniques are important because they are broadly applicable to any text content, for example posts made on social media sites such as Facebook or Twitter. They can also be applied to similar tasks, such as predicting the number of stars given by a Yelp review.

Over the years, there have been various approaches towards solving such tasks. Traditional approaches are often based on the Bag of Words model, where the frequency of each word in a piece of text is used as a feature for training a classifier or regressor [2]. Another approach, introduced in a work called VADER, uses human-validated sentiment values for each word to construct overall sentiment values for a sequence of text [5]. Currently, state of the art approaches typically utilize deep learning [3]. One such modern work is BERT, a deep learning transformer model that significantly boosted accuracy in various NLP tasks [4].

2 Related Work

Our approach will focus on enhancing BERT. Here we discuss two works that focus on building upon existing deep learning models to improve their performance.

2.1 How to Fine-Tune BERT for Text Classification?

This work by Sun et al. [3] evaluates various fine-tuning strategies for BERT. The work finds that within-task and in-domain pre-training are the most helpful strategies, which informed our decision to base our approach on fine-tuning BERT. Additionally, this work finds that when meeting BERT's 512-token limit, taking 128 tokens from the beginning and 382 tokens from the end of the text results in the lowest error rates, followed by taking 510 from the end, then 510 from the beginning. This helped inform our approach as we needed to first truncate text to meet BERT's limit, and later decided further truncate text to reduce training times.

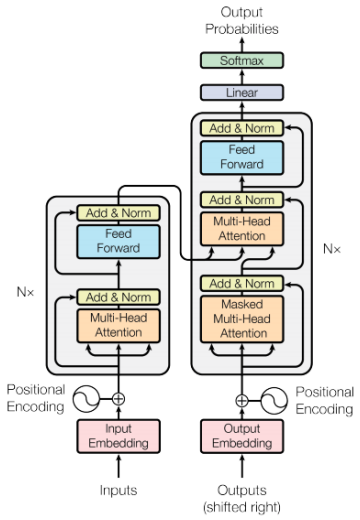

Figure 1: Transformer model architecture [7].

2.2 Enhancing deep learning sentiment analysis with ensemble techniques in social applications

This work by Araque et al. [6] discusses the benefits of ensembling to Sentiment Analysis. It includes feature ensembling, where different feature construction methods are combined, and classifier ensembling, where the predictions of multiple different classifiers are combined. The work concludes that combining different types of sentiment information, for example hand-crafted features with automatically extracted text embeddings, or combining multiple weaker predictors, can significantly improve sentiment classification performance relative to the baseline. This informed our approach when ensembling BERT with other models that were weaker, but might provide additional information.

3 Background

Transformers were introduced by Vaswani et al. in "Attention is All you Need" as a solely attention-based alternative to existing sequence modeling approaches [7]. The basic transformer model architecture, shown in Figure 1, uses an encoder-decoder approach with several multi-head self-attention layers and fully connected layers. In self-attention, each token is directly connected to all tokens (in the encoder) or all preceding tokens (in the decoder) through key-value querying, helping solve some of the issues with forgetting associated with traditional recurrent neural networks (RNNs) or long short-term memory (LSTM) models. Additionally, because the encoder and decoder do not require tokens to be processed in sequence, they can be parallelized, allowing faster computation [7].

BERT is a transformer-based model that further improves upon this approach by using just an encoder. Instead of decoding outputs, it allows self-attention to all other tokens, regardless of position, and masks random input tokens during training to build up contextual understanding [4]. It is pre-trained with a multi-billion word text corpus, and can be fine-tuned to the desired task.

4 Methodology

Our approach seeks to leverage the power of BERT, while balancing it with our computational limitations. We started with a baseline of BERT and experimented with various novel ideas to improve upon our results.

4.1 Multiple Loss Functions

For this task, we want to have both a high accuracy and a low average star error. To account for this, we train on two separate loss functions—cross-entropy loss, to accurately classify a review into the five possible star values, and L1 loss, to reduce the distance between the model's prediction and the true star value. Thus, our combined loss looks like:

$$\mathcal{L}(w, x, y) = \frac{1}{N}\left((1-\alpha)\sum_{i=1}^{N} -log\left(\frac{e^{w_i[y]}}{\sum_j e^{w_i[j]}}\right) + \alpha\sum_{i=1}^{N}|x_i - y_i|\right),$$

where α is a hyperparameter corresponding to the weight of the L1 loss and $1 - \alpha$ is the respective cross-entropy loss. The inputs w and x are two separate outputs from BERT. w is created by passing BERT's final hidden layer activations through a linear layer of size [hidden_state x 5], while x is created similarly except the linearly is was of size [hidden_state x 1]. y is the correct rating of a review.

We present two different methods to combine the two losses when predicting review star values. The first approach is to take the softmax of the five logits output by the classification head, and add α (0.2 worked well) to the value indexed by the star value nearest to the regression head's output. This serves as a voting method where the regression head can help break ties when there are multiple logits with similar values. This allows our model to tend towards the prediction with a lower star error when uncertain what prediction to make. The second approach is to "spread out" the bonus applied by the regression head between nearby logits, by adding a term that decreases with the distance from the regression head output (the exact formula used is below). This helps account for uncertainty in the regression head when its output is between two different star values.

For each logit l_i, $i \in \{1, ..., 5\}$ from our classification head and with a regression head output r, with the two schemes our final predictions are:

$$\underset{i}{\operatorname{argmax}} \left[l_i + \alpha * \mathbb{1}\{\lfloor r \rceil = i\} \right] \tag{1}$$

$$\underset{i}{\operatorname{argmax}} \left[l_i + \frac{\alpha}{2^{|r-i|}} \right] \tag{2}$$

Note: For a value $\lfloor i \rceil$, i is being rounded to the nearest integer.

4.2 DistilBERT Ensembling

Due to the small size of our data set, BERT over-fits to our data set fairly quickly. To combat this, we devise an ensemble approach where K models of BERT are trained on a random partition of the data and each model has a weight, w_i, determined by its validation accuracy a_i where:

$$w_i = \left\lfloor \frac{100 * e^{a_i}}{e^{a_i}} \right\rfloor$$

We use this new weight in a voting system where each model's weight is its number of votes. In this system, we take the mode of the votes as our new prediction. By doing ensemble learning we are able to reduce the variance and create a more generalized model.

Because of the need to train multiple BERT models, we used DistilBERT for this part in order to reduce computational load. DistilBERT is a variation of BERT that is 60% faster while retaining 97% of accuracy of NLP tasks through knowledge distillation [8]. Also, due to the limited time for experimentation, we ran each model through around 20% of the training data available.

4.3 BERT+VADER Ensembling

The pre-trained BERT model is trained primarily on a Wikipedia data set, which may introduce bias as the data varies greatly in diction and style when compared with the Yelp review data [4]. To compensate for this, we propose ensembling BERT with VADER, since it has a "gold-standard sentiment lexicon that is especially attuned for microblog-like contexts" [5]. We hypothesize that by combining the outputs of BERT and VADER through a multilayer perceptron (MLP), the bias of the model will decrease, potentially leading to better results. Our MLP is trained on both classification and regression head outputs of BERT, along with the three sentiment values provided by VADER (positive, neutral, and negative), for a total of 9 input features.

5 Results

Since our enhancements are centered around BERT, we first trained a baseline model on BERT alone. It gave us a validation set accuracy of around 78%. Initially, ensembling the regressor and classifier heads resulted in a boost of around 1.2 percentage points to validation set accuracy, as shown in Table 1. After enabling the second prediction scheme for the regressor and classifier ensembling, we saw an additional boost of around 0.5 percentage points. We hypothesize that the dual-head scheme improves the generalizability of the model by acting similarly to two separate models, while leveraging the power of shared parameters.

Next, ensembling BERT with VADER also provided an initial improvement of around 1.2 percentage points over the baseline. This seems to confirm that adding human-validated sentiment information to BERT can be helpful. Future

Table 1: Experiment results (128 tokens/review)

Description	Peak validation accuracy (%)	MAE (stars)
Base BERT	78.22	0.2781
Classifier+regressor ensemble (scheme 1)	79.38	0.2527
Classifier+regressor ensemble (scheme 2)	79.89	0.2399
BERT+VADER ensemble (MLP)	79.40	0.2564
5 DistilBERT ensemble (voting)	76.27	0.3501

Table 2: Experiment results (Base BERT)

No. of tokens	Peak validation accuracy (%)	MAE (stars)
60	76.03	0.3292
128	78.22	0.2781
256	79.70	0.2357

Table 3: Experiment results (Base BERT, Classifier+regressor scheme 1)

Description	Peak validation accuracy (%)	MAE (stars)
128 tokens from beginning	79.38	0.2527
64 tokens from beginning + 64 tokens from end	79.69	0.2470
128 tokens from end	79.30	0.2527

Table 4: Experiment results (Base BERT, 60 tokens)

Batch size	Peak validation accuracy (%)	MAE (stars)
10	76.03	0.3292
32	65.27	0.7266

work here could include varying the size of the BERT output and/or training it differently, changing the architecture of the neural network with which BERT and VADER are combined, or using a different type of model entirely.

The ensemble of 5 DistilBERTs was not as successful as the previous two approaches, and resulted in a lower accuracy than base BERT. We suspect this was due to the reduced size of the data set each DistilBERT instance was trained on, in addition to DistilBERT's inherently reduced power compared to BERT. In future work, we could iterate on this idea by adjusting the amount of training data, changing the way the different models vote, or adjusting other hyperparameters like the number of models.

Another experiment we carried out (Table 2) was adjusting the maximum length of inputs to BERT. Due to hardware and time constraints, we stopped at 256 tokens, but found that increasing the number of tokens consistently improved accuracy. We also found that using the beginning and end of a review performed better than using just the beginning or end (Table 3), and that using a smaller batch size resulted in much better performance (Table 4).

By combining the best configurations from the above results, we achieved a peak of 80.9% accuracy and 0.219 mean absolute error (MAE) on our validation set, as well as 80.6% accuracy and 0.226 MAE on our withheld test set.

6 Conclusion

In this report, we discuss various methods of fitting, enhancing, and ensembling BERT in order to improve its accuracy in predicting Yelp review ratings. We reduce bias by ensembling BERT with VADER, and present preliminary work on ensembling multiple BERT models in order to reduce variance and thereby reduce overfitting. We also devise two novel prediction schemes using both classification and prediction heads that improve both accuracy and mean absolute error significantly. Through these methods and a comprehensive hyperparameter exploration, we improve accuracy by 3.5% and mean absolute error by 27% relative to our initial fine-tune of the already powerful BERT. Finally, there are many potential avenues for further investigation, for example additional hyperparameter tuning on our ensemble of DistilBERTs, and the potential improvements to our BERT+VADER scheme discussed above.

7 Team Contributions

Charles: 50%. I wrote the code for training the initial BERT model. We pair programmed our novel implementations. We each wrote different sections of the report.

Kevin: 50%. I researched and implemented ensemble techniques. I performed all of the training and testing and optimized the code to the specifications of my machine.

References

[1] Liu, Bing. "Sentiment analysis: Mining opinions, sentiments, and emotions." *Cambridge University Press*, 2015.

[2] Pang, Bo et al. “Thumbs up? Sentiment Classification using Machine Learning Techniques.” *Proceedings of EMNLP*, 2002, pp. 79–86.

[3] Sun, Chi et al. “How to Fine-Tune BERT for Text Classification?” *China National Conference on Computational Linguistics (CCL)*, 2020.

[4] Devlin, Jacob et al. “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.” *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*, 2019.

[5] Hutto, C.J. et al. "VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text." *Proceedings of the Eighth International AAAI Conference on Weblogs and Social Media*, 2015.

[6] Araque, Oscar et al. "Enhancing deep learning sentiment analysis with ensemble techniques in social applications." *Expert Systems with Applications*, Volume 77, pp.236-246, 2017.

[7] Vaswani, Ashish et al. "Attention is All you Need." *31st Conference on Neural Information Processing Systems (NIPS)*, 2017.

[8] Sanh, Victor et al. "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter." *ArXiv Preprint*, 2019.